面向组件接口的XACML变异测试策略

聂 南[1,2]，夏启明[1]，姚俊峰[1]，何克清[1]

(1. 武汉大学软件工程国家重点实验室 武汉 430074; 2. 郑州轻工业学院计算机与通信学院 郑州 450002)

【摘要】XACML是一种适用于各种信息资源保护的访问控制语言。由于其严格的语法规范，且具有多种平台之间的可移植性，非常适用于各种组件交互的安全管理。借助该访问控制语言提出的一种面向组件的三层访问控制方法，组件交互、接口调用和参数访问都能实现安全控制。在该基础上设计了相应的变异测试策略，规则变异可以导致策略变异，策略变异可以导致整个策略集的变异；反之亦然。通过测试具体实例与验证其语义模型，该测试策略为组件访问及交互提供了安全保障。

关 键 词 访问控制接口; 组件; 变异测试; 访问控制语音

中图分类号 TP311 **文献标识码** A doi: 10. 3969/j. issn. 1001-0548. 2009. 02. 31

Component Interface -Oriented XACML Mutation Testing Policies

NIE Nan[1,2], XIA Qi-ming[1], YAO Jun-feng[1], and HE Ke-qing[1]

(1. State Key Laboratory of Software Engineering, Wuhan, University Wuhan 430072;

2. School of Computer and Communication, Zhengzhou University of Light Industry Zhengzhou 450002)

Abstract A kind of three level access control policy towards component is presented by extensible access control markup language (XACML) for the protection of component interaction, interface invocation, and parameters access. Based on this policy, the mutation test strategies are designed: policy mutations follow policy mutations, policyset mutations follow policy mutations, and vice versa. Both the case study and semantical verification shows that the access control of component interface and interactions can be tested by XACML mutation policies.

Key words access control interface; component; mutation test; XACML

组件的访问安全涉及组件的内外部接口和组件的交互，因此是一个需要综合考虑的问题。目前提出的测试方法，特别是接口测试方法，如边界值分析、划分等价类，以及在此基础上的接口参数组合测试[1]等多是面向程序的结构，而不是针对接口访问信息的保护。其缺点在于不能针对接口的安全问题进行检测，缺乏统一的描述机制，而且开发平台不能适用于多种平台下的组件。

访问控制是一种实现计算机资源安全管理的方法，2003年OASIS(organization for the advancement of structured information standards)制定了基于XML的访问控制策略和访问控制请求/响应描述语言规范——XACML。对于组件接口，可以充分利用XACML对组件实现访问控制。此外，为确保访问控制准确、全面的实施，还需要进行相关的测试。文献[2]研究了基于XACML的变异测试方法与模型，提出并且设计了测试用例生成工具。文献[3]研究了访问控制的自动测试方法，扩展了GFAC测试接口，并且使用该方法实现了在Linux+RSBAC环境下对自主访问控制的自动测试。文献[4]引入作用于组件接口及内部数据集的XML Schema变异算子，提出一种基于XML API的组件接口扩展变异测试方法。本文基于XACML的特别功能效果，提出了面向组件接口的扩展访问策略，实现接口访问控制的变异测试方法。该方法继承了原方法的多种优点，如测试框架与组件耦合性好、通用性强、可跨平台等。而且该策略针对组件接口访问控制的问题，具有更加迫切和实际的应用价值。

1 组件及其接口的访问控制

1.1 组件及其接口扩展定义

目前，组件尚没有一个统一且标准的定义。文献[5]将其定义为一个近乎独立的、可替换的、封装的，为实现某些功能的数据和操作模块，可看作由

收稿日期：2007 – 11 – 13；修回日期：2008 – 09 – 18

基金项目：国家重点基础研究发展计划(2006CB708302)；中小企业公共技术服务机构补助资金(08C26244202133)

作者简介：聂 南(1973 –)，男，博士，主要从事软件工程与理论方面的研究.

接口和内部规范组成的。接口包括一组说明组件与外部环境交互行为的端口描述和说明组件功能或相关性质的描述。文献[4]对其接口进行扩展，并且给出了相关定义，在此基础上，可以开发组件的访问控制扩展接口，通过XACML与组件的集成，从而实现组件的访问控制。

1.2　组件及其接口的访问控制

定义 1　一个组件访问控制系统C是一个4元组<*U*, *S*, *P*, *R*>。*U* 表示访问组件的用户集合；*P*为用户与资源映射的集合；*R*为映射时采用的规则；*U*={administrator, user, developer，特定用户等}；*S*={(数据集, 方法集, 参数集)(个数, 大小, 范围等)}; *P*={*U X S*} 包括不同用户对于不同资源访问控制的影射约束(时间, 次数, 角色等); *R*={只读, 可写, 浏览, 允许, 拒绝等}。

性质如下：

(1) $U^R \rightarrow S$； $U \in$ admin *R*= WRITE|READ|APPEND|PERMIT|DENY|…can-restrict(*U*,*r*)(Role (*u*) = admin, Restrict(*r*)= (time=No, access number=No)，对于超级用户的约束都约定为否。

(2) $U^R \rightarrow S$； $U \in$ user *R*=WRITE| READ | PERMIT|DENY； can-restrict(*U*,*r*)(Role(*u*)= user, Restrict(*r*)= (time=Yes, access number=No)， 对于普通用户可以设置部分约束。

(3) $U^R \rightarrow S$； $U \in$ guest *R*=READ|DENY；can-restrict(*U*,*r*)(Role(*u*)=guest, Restrict(*r*)= (time=Yes, access number=Yes) ，对于访客可以设置更严格的约束，使其只具有浏览权限。

此外，还可以根据组件系统具体规定，定义特定用户对资源的访问控制及约束条件。

1.3　XACML实现访问控制

XACML已经发展为一种可以描述授权信息的统一的策略描述语言，为多粒度访问控制提供了语言基础。在XACML中访问请求描述的自然语义为“在当前条件下，主体以某种方式访问资源”。其中条件、主体、方式、资源通过属性值来描述，即通过<XACML：Environment> 描述条件；<XACML：Subject>描述主体；<XACML：Resource>描述资源；<XACML：Action>来描述访问方式。使用XACML中定义的元素，可以将请求形式化地描述为*R*(Subject, Resource, Actio, Environment)。其能够对访问控制策略和访问控制请求/响应产生过程加以描述，可以根据主体、资源、环境的属性以及所采取的行为进行控制——允许还是拒绝。实际上，返回的结果有4种：允许(permit)、拒绝(deny)、无法决定(indeterminate)和不适用(notapplicable)。XACML不仅提供了一系列逻辑算法对整个授权过程进行控制，而且提供了支持定义新功能、数据结构、合成逻辑算法等标准可扩展点[6]。借助XACML的访问控制定义，本文提出面向组件接口的扩展访问测试策略，给出以下组件接口访问控制和测试框架以及形式化用例。

2　面向组件的XACML变异测试

XACML将主要解决以下组件接口访问问题：

(1) 对组件交互提供粗粒度的访问约束。

(2) 创建一种可移植的、标准的方式来描述访问组件接口及其属性。

(3) 提供一种机制，以比简单地拒绝访问或授权访问更细粒度的控制访问，即在“允许”或“拒绝”之前或之后执行某些操作。

2.1　XACML变异测试框架

面向组件接口的XACML变异测试体系结构如图1所示。

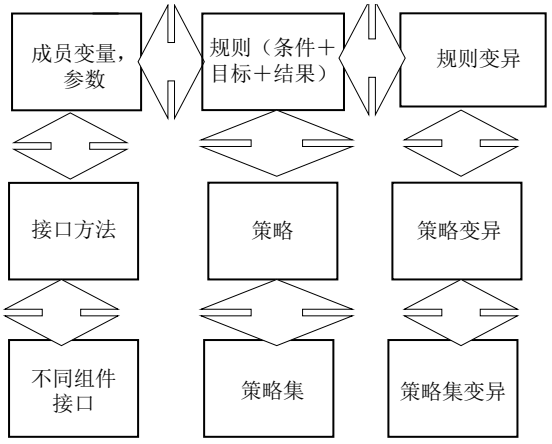

图1　组件访问控制测试结构图

基于图1，本文提出一种三层规则变异推导方法：

定义 2　Policyset变异(P,目标,义务)<—> Policy变异(一组规则Rule、规则组合算法的标识符、一组义务和一个目标) <—>Rule变异(条件(Indeterminate, False, True)、结果(Permit, Deny)和目标)。

括号内可以看成是变异项，它们之间是*x*的关系，如Rule变异的个数为：条件×结果×目标。规则变异的改变可以导致Policy变异的改变，某Policy变异的改变可以导致Policyset变异的改变；反之Policyset目标逻辑的改变，导致Policy目标组合逻辑的改变；Policy目标的改变导致规则结果的逻辑改变。该规则推导方法可适用于以下的组件三层变异

测试。

2.2 组件三层变异测试

组件三层变异测试主要包括以下3项内容：

1) 组件交互级别的变异测试，如系统的登录组件和数据库组件的交互等。

组件交互的测试可以看成是确保系统级别的安全性。组件的交互主要是由组件接口调用来实现的。接口由一组角色定义，每个角色说明参与交互的组件在这一交互中所应有的行为。

基于XML的某WEB组件交互访问控制测试模型(access control test model, ACTM)为：ACTM(R,M, Σ)，其中R是一个接口角色定义集；M是组件交互信息对，包括：(访问请求，访问响应)；Σ是访问变异算子集合。由R发出，M完成访问请求与响应，然后Σ的变异算子可以进行变异，实现请求与响应的测试与验证。

2) 组件接口级别的测试，包括对组件接口函数调用的访问，需要对访问控制方法进行管理和约束。具体包括以下定义：

$U_m \rightarrow S$；$U \in$ admin can-access(all methods)

$U_m \rightarrow S$；$U \in$ user can-access(write methods)

$U_m \rightarrow S$；$U \in$ guest can-access(read methods)

必要时可建立函数调用访问控制列表，不同类型的用户，对于函数有不同的调用级别。

如某函数的用户访问时间约束定义为Time$R(u)$=C(condition)，其中R表示注册函数；u是USER(u)中用户类型，采用时间条件约束各种用户对函数的调用；C表示约束条件。定义的变异项包括：用户、时间、函数等。

3) 函数参数访问变异。接口函数包括许多类型及精度不同的参数，对于它们的访问控制，可以给出以下定义：

(1) Access_field(parameters)：参数精度的访问约束，如整形参数可设置约束长度–10 000～+10 000，否则参数请求将被拒绝。

(2) Access_type(parameters)：参数类型的访问约束，如要求为双精度，如果传递浮点型，访问将不被允许。

(3) 1&&2 ：约束条件的组合，必须满足1和2条件的组合。如满足以上(1)、(2)组合条件。

可以定义的变异项包括：用户、参数、约束条件等。可确保在预期的安全性情况下，用户只能访问特定的功能或用例，或者只能访问有限的数据。如可能允许所有人输入数据、创建新账户，但只有管理员才能删除该数据或账户。如果具有数据级别的安全性，测试就可确保“用户类型一”能够看到所有菜单消息，而“用户二”只能看见部分的统计数据。

3 实例研究

本文已将XACML函数库与JAVA BEAN 进行了组合设计，在WINDOW XP+ECLIPSE 3.1.1/ UBUNTU7.1+ECLIPSE 3.2.0平台上进行了实例研究。由于生成的.class可以由JAVA虚拟机在不同平台上运行，充分体现了该设计方法的跨平台性。但是由于已知的XACML函数库比较复杂，还存在若干访问组件的架构搭建的问题，某些具体技术还需要进一步探索和发展才能更好地实现XACML与组件的组合。为实现基于XACML的组件的访问控制，目前，文献[7]已推出支持XACML的函数库(SUN XACML)。在此基础上推出LINUX环境的软件包rw-XACML，借助组件的扩展接口，能够实现面向组件的跨平台访问控制功能。

以下实例采用XACML policy实现对某组件接口的控制，部分内容取自文献[8]，并且在此基础上实现变异测试。变异体主要是条件、目标、结果逻辑变量值及它们的数据类型。

```
<?xml version="1.0" encoding="utf-8"?>
<PolicySet PolicySetId="PolicySet1"
<Policy … RuleCombiningAlgId="变异体">
  <Target>  <Subjects> 变异体  </Subjects>
  <Resources> 变异体  </Resources>
  <Actions> 变异体 </Actions>  </Target>
  <Rule
RuleId="urn:oasis:names:tc:XACML:1.0:conformance-test:IIA1:rule" Effect="变异体">
<Condition FunctionId="变异体"> <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">变异体</AttributeValue>
  //该变异体逻辑决定response的decision逻辑
<AttributeSelector
RequestContextPath="//context:ResourceContent/db:user/db:view_once/text()" DataType="http://www.w3.org/ 2001/XMLSchema# string"/> </Condition>
<Target> <Subjects> <Subject><SubjectMatch MatchId= "变异体"> <AttributeValue DataType="变异体">变异体</AttributeValue>//不同的用户
<SubjectAttributeDesignator
```

AttributeId="urn:oasis:names:tc:XACML:1.0:subject:subject-id" DataType="变异体" SubjectCategory="urn:oasis:names:tc:XACML:1.0:subject-category:access-subject" MustBePresent="变异体" /> </SubjectMatch> </Subject> </Subjects><Resources> <Resource> // … 同理省略 </Resource> </Resources> <Actions> //同理省略
</Actions> </Target> </Rule> </Policy>
<Policy … RuleCombiningAlgId="变异体"> … </Policy>…</PolicySet>

request申请例子如下：

<?xml version="1.0" encoding="utf-8"?>
<Request <Subject> <Attribute AttributeId= "urn:oasis:names:tc:XACML:1.0:subject:subject-id" DataType="变异体">
<AttributeValue> user
</AttributeValue>//用户变异体
</Attribute> </Subject>
<Resource>…同理省略 </Resource>
<Action>…同理省略</Action>
</Request>

得到的Response结果：

<?xml version="1.0" encoding="utf-8" ?>
–<Response xmlns="urn:oasis:names:tc:XACML:1.0:context">
–<Result><Decision> 变异结果 </Decision> </Result>// 4种结果(Permit)、(Deny)、(Indeterminate)和(NotApplicable)
–<Status> <StatusCode Value="变异结果" /> </Status> </Response>

借助Policies 和Response中的变异体，通过测试多个类型用例的Request/Response数据，可以得出多种访问控制结果，从而测试出对受保护资源的限制结果。可确保只有具备系统访问权限的用户能够依据一些具体的要求访问组件应用系统。

4 结 论

本文借助XACML访问控制语言，提出一种面向组件的三层访问控制策略，在此基础上进行了变异测试。借助具体实例表明，该测试策略为组件参数、接口及组件交互提供了访问安全保障。还能够结合模型检验技术，转化XACML为自动机建模工具的输入语言，进而实现接口安全测试的形式化验证[9]。由于XACML已经成为多种语言平台的访问控制标准，因此该方法可以应用于Linux/Windows等多种操作系统的测试环境，从而具有跨平台性等多种优点。借助XACML实现的组件测试策略，同样能够被应用到其他信息资源的安全管理，因此具有深远而广阔的应用前景。

参 考 文 献

[1] 聂长海, 徐宝文. 基于接口参数的黑箱测试用例自动生成算法[J]. 计算机学报, 2004, 27(3): 382-388.
NIE Chang-hai, XU Bao-wen. An automatic black box test case generation algorithm based on interface parameters[J]. Chinese Journal of Computers, 2004, 27(3): 382-388.

[2] MARTIN E, Xie Tao. A fault model and mutation testing of access control policies[C]//Proc 11th International Conference on World Wide Web. New York: ACM, 2007, 667-676.

[3] 丁洪达, 曾庆凯, 包必显. 访问控制的验证测试方法研究[J]. 计算机工程, 2007, 33(1): 161-163.
DING Hong-da, ZENG Qing-kai, BAO Bi-xian. Study of test approach on access control[J]. Computer Engineering, 2007, 33(1): 161-163.

[4] 聂 南, 谢晓东, 甘 勇, 等. 基于XML API的组件扩展接口变异测试方法[J]. 计算机科学, 2008, 35(6): 283- 287.
NIE Nan, XIE Xiao-dong, GAN Yong, et al. Extension Interface Mutation Testing for Component Based on XML API[J]. Computer Science, 2008, 35(6): 283-287.

[5] 毛澄映, 卢炎生. 构件软件回归测试用例选择策略[J]. 计算机研究与发展, 2006, 43(10): 1767-1774.
MAO Cheng-ying, LU Yan-sheng. Strategies of regression test case selection for component-based software[J]. Journal of Computer Research and Development, 2006, 43(10): 1767-1774.

[6] OASIS. XACML v3.0 Administration Policy Version 1.0 working draft[J/OL]. [2009-1-1]. http://www.oasis-open.org/committees/download.php/25635.

[7] SUN. Sun’s XACML implementation[J/OL]. [2006-5-1]. http:// sunXACML. sourceforge.net/.

[8] FEDOEA. Fedora Authorization with XACML Policy Enforcement[J/OL]. [2007-10-1]. http://fedora.info/download/2.1b/userdocs/server/security/AuthorizationXACML.htm.

[9] ZHANG Nan, RYAN M D, GUELEV D P. Synthesising verified access control systems through model checking[J]. Journal of Computer Security, 2008, 16(1): 1-61.

编 辑 张 俊